Graph Neural Networks for Efficient AC Power Flow Prediction in Power Grids

Amir Talebi
Electrical and Computer Engineering
North Carolina State University
Raleigh, NC, USA
amirtalebi879@gmail.com

Kaixiong Zhou
Electrical and Computer Engineering
North Carolina State University
Raleigh, NC, USA
kzhou22@ncsu.edu

Abstract

1 Introduction

The Optimal Power Flow (OPF) problem is fundamental and one of the most important optimization in power system operations [1]. It involved the minimization of generation costs while meeting load demands and satisfying the various grid constraints such as generator capacities (Active and reactive power limitsy), voltage magnitude and phase limits and line thermal limits. Efficient and accurate solutions to the OPF problem are essential for real-time power system management, especially as modern power systems become increasingly complex with the integration of variable renewable energy sources and increase of the number of nodes.

Traditional OPF solvers, such as Newton-Raphson or Interior Point Methods (IPOPT), often face challenges in scalability and computational efficiency. As the number of nodes (buses) and edges (transmission lines) in the network increases, the computational complexities grows significantly and make real-time applications difficult. These methods also tend to be computationally intensive, limiting their use in large-scale grids or in systems with high variability introduced by renewable energy sources. The exact formulation of the problem is commonly referred to as ACOPF and the approximation as DCOPF. Incorporating AC dynamics results in a non-convex nature of OPF problem due to nonlinearities in the power flow equations

Recent advancements in Machine learning and deep learning,motivated by the possibility of accurately generating large amounts of data, and particularly Graph Neural Networks, offer a promising solutions to address these challenges. GNNs are designed to handle graph-structured data effectively which make them well-suited for modeling power grids where buses can be represented as nodes and transmission lines as edges. By leveraging GNNs, we aim to develop a model that can predict OPF solutions efficiently, maintaining high accuracy while significantly reducing computation time.

In this project, we propose a novel approach using GNNs to improve the accuracy and scalability of OPF predictions. We will explore multiple GNN topologies, such as multi-hop message-passing GNNs, which allow for deeper propagation of information across the grid, and attention-based GNNs that assign varying importance to different buses in the network based on their influence on OPF solutions. The model will be trained on standard IEEE test cases, and its performance will be compared against traditional solvers, including Newton-Raphson and IPOPT. The objective is to develop a model that scales well to large power grids, respects operational constraints, and is capable of real-time grid management.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

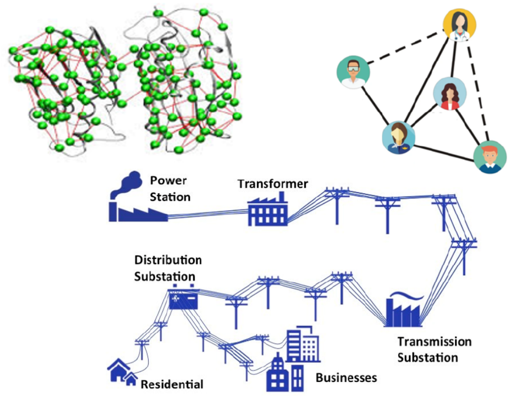

Figure 1: Proteins [2], social networks [3], and electrical power grids [4] are all graphs

2 Related Work

In recent years, machine learning (ML) approaches have been introduced to approximate or solve OPF more efficiently. Among these, multi-layer perceptrons (MLPs) were some of the earliest ML models applied to power system problems. Studies such as [5] applied fully connected network (MLP) to imitate the output of ACOPF and predict power flow variables and generator outputs, However, MLPs face significant challenges with local minima and overfitting, especially when applied to large-scale power systems with non-linearities. They also struggle to model the graph structure of the power grid, where buses (nodes) and transmission lines (edges) are interconnected in a highly structured manner. The failure to leverage this topological information leads to inaccurate result and poor performance, especially in generalization across different grid topologies.

To address these shortcomings, Graph Neural Networks (GNNs) have emerged as a more suitable architecture for power system applications. GNNs excel in handling graph-structured data which make them suitable for power grids, where buses and transmission lines naturally form a graph. GNNs use a message-passing mechanism, allowing each node to aggregate information from neighboring nodes, which mirrors the flow of electrical voltages and currents between interconnected buses in power system.

Authors in [6] were among the first to apply GNNs to the OPF problem. They developed a GNN-based model that uses imitation learning to predict OPF solutions based on the outputs of traditional solvers like IPOPT. The model demonstrated significant improvements in computation time compared to traditional solvers while maintaining high accuracy, particularly for the IEEE-30 and IEEE-118 bus systems. This approach successfully exploits the grid topology to improve efficiency and provided a foundation for future research in using GNNs for OPF.

[7] proposed a Topology-Aware Graph Neural Network that incorporates both the spatial structure of the grid and physical constraints into the learning process. Their model introduces AC-feasibility regularization and ensured that the GNN's predictions commit to the physical laws of power flow, Kirchhoff's laws and generator limits. This method enhanced the generalizability of GNN models across different grid topologies and more importantly ensures that solutions are physically feasible, a critical requirement in real-world power systems.

Further extending the capabilities of GNNs, [8] introduced a Typed Graph Neural Network (TGNN) approach in their work on power flow analysis. TGNNs differentiate between different types of buses (e.g., generator buses, load buses) in the power grid, allowing the model to treat each node type according to its operational role. This added complexity in node classification improves the accuracy of power flow predictions, especially in cases with diverse bus types and heterogeneous grid configurations. Their experiments with typed GNNs showed better performance compared to standard GNN models, particularly in grids with mixed bus types.

Authors in [9] provide a comprehensive review of the application of GNNs in power systems, highlighting their advantages in terms of scalability, generalizability, and performance when applied to complex, non-linear power system problems like OPF. The authors emphasize the ability of GNNs to handle dynamic grid configurations and temporal variations which make them effective in environments with high renewable penetration and fluctuating load conditions. Their review also covers various GNN architectures, including ChebNet and spectral-based GCNs, which have been applied to solve power flow and fault detection problems. The study's key contribution is its focus on task analysis, where GNNs outperform conventional deep learning models by leveraging the intrinsic graph structure of the power grid. They also discuss the critical challenges of data availability and the need for advanced regularization techniques to ensure physical feasibility in GNN predictions.

[10] introduced a model that combines Proximal Policy Optimization (PPO) with GNNs for OPF. Their work leverages the decision-making capabilities of PPO to control generator outputs in a power grid while using GNNs to model the spatial relationships within the grid. Their experiments showed that this approach outperforms traditional solvers like DCOPF in both cost minimization and constraint satisfaction, particularly in dynamic environments where grid topology or load conditions change.

Additionally, [11] explored the application of Graph Neural Solvers (GNS) to directly solve power flow equations by minimizing violations of Kirchhoff's laws. Their work focused on developing a graph-based solver that could scale efficiently with grid size while ensuring commitment to physical laws, laying the groundwork for GNN-based models that solve power flow equations directly without the need for traditional optimization solvers.

More recently, [12] introduced a comparative study between Koopman Operator-based models and GNNs for learning power grid dynamics. This work highlighted the advantages of GNNs in capturing the spatio-temporal dynamics of power grids, demonstrating their utility not only for static optimization problems like OPF but also for transient analysis and fault detection. The study showed that GNNs could outperform Koopman models in terms of generalizability and adaptability to changing grid conditions.

Incorporating more recent work, [13] proposed a Graph Attention Network (GAT) that integrates physical constraints such as Kirchhoff's laws. This hybrid model focuses on incorporating domain-specific physics into the learning process, ensuring both scalability and physical accuracy. The GAT model allows for efficient propagation of information across nodes (buses), addressing scalability concerns while preserving physical feasibility. This makes it highly suitable for real-time OPF applications in complex, renewable-integrated grids.

[14] demonstrate the use of GNNs in a probabilistic context. Their focus is on using attention mechanisms to prioritize critical nodes in a grid, especially under uncertain conditions caused by renewable energy sources. This marks a shift from deterministic to probabilistic approaches in power flow and adress the challenges posed by stochastic variations in renewable energy generation.

The progression from traditional solvers and MLPs to GNNs represents a shift towards more scalable and efficient OPF solutions. The key advantage of GNNs lies in their ability to incorporate the grid topology which improve both prediction accuracy and computational efficiency. However, challenges remain, particularly in ensuring that GNNs can handle real-time grid operations and adapt to dynamic topologies in a computationally feasible manner.

3 Methodology

3.1 AC Power Flow Problem Formulation

AC Power Flow analysis aims to determine the voltage magnitudes (V) and phase angles (δ) at each bus in a power grid, while ensuring active (P) and reactive (Q) power balance across the network. The nonlinear power flow equations are given by:

$$P_i = V_i \sum_{j=1}^{n} V_j (G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)),$$

$$Q_i = V_i \sum_{j=1}^{n} V_j (G_{ij} \sin(\delta_i - \delta_j) - B_{ij} \cos(\delta_i - \delta_j)),$$

where P_i and Q_i represent the net real and reactive power injections at bus i, V_i and δ_i are the voltage magnitude and angle, and G_{ij} and B_{ij} are the real and imaginary components of the admittance matrix.

The primary goal of solving these equations is to predict the voltage profile of the system under different load conditions, which forms the basis for the Optimal Power Flow (OPF) solution.

The novelty of our approach lies in updating the voltage magnitude (V) and phase angle (δ) as dynamic node features during training, which is a critical step in accurately solving the AC Power Flow problem. Additionally, the bus types (PV, PQ, Slack) were included as node features to capture the specific characteristics of each bus type.

3.2 Dataset Generation

The datasets for our experiments were generated using the standard IEEE test systems (14-bus, 30-bus, and 57-bus) with realistic %40 variations in load demands. The main objective was to create diverse scenarios that effectively evaluate the robustness and generalization capability of our GNN models. To achieve this, we employed pandapower, a Python-based library tailored for power system analysis and leveraged its built-in IEEE test cases and power flow capabilities. The load variations were designed to simulate both daily and seasonal patterns and incorporate realistic fluctuations in load values. This network includes predefined bus configurations, loads, generators, and transmission lines which is an ideal test case for load flow analysis. The script generates 10 separate datasets, each containing 10000 samples, which restuls in a total of 100,000 data points per bus system configuration.

- **IEEE 14-Bus System:**
  - A small and simple transmission network containing 14 buses, 5 generators (including the slack bus), 11 loads, and 20 branches.
  - Commonly used for educational purposes and serves as an excellent starting point for testing basic load flow algorithms. Its smaller network size allows for quick evaluation of the model's fundamental capabilities before scaling up to larger systems.
  - Tests the GNN's ability to handle fundamental power flow scenarios, including handling different bus types (Slack, PV, PQ), and accurately predicting voltage magnitudes and angles under various load conditions.
- **IEEE 30-Bus System:**
  - A medium-sized network with 30 buses, 6 generators, 21 loads, and 41 branches.
  - Widely used in the literature for testing load flow and Optimal Power Flow (OPF) algorithms due to its increased complexity. The diverse mix of generation and load configurations makes it an excellent test case for evaluating the scalability and generalization capabilities of our GNN models.
  - Provides realistic scenarios involving larger variations in load and generator setpoints, helping us analyze the model's response to typical operational constraints.
- **IEEE 57-Bus System:**
  - A more complex transmission network containing 57 buses, 7 generators, 42 loads, and 80 branches.
  - Commonly used for load flow analysis, contingency analysis, and testing the robustness of power flow solvers. It poses a challenging scenario due to its larger network size and increased interconnections, requiring the GNN model to effectively propagate information across a wider spatial area.
  - Helps evaluate the model's ability to handle medium to large-sized systems and test its robustness in predicting power flow variables accurately under diverse operating conditions.
- **IEEE 118-Bus System:**
  - A large and realistic transmission network with 118 buses, 19 generators, 99 loads, and 186 branches.
  - Widely used for studies involving OPF, contingency analysis, and voltage stability. It provides a comprehensive test for scalability, allowing us to validate the generalization capability of our model on real-world, large-scale power systems.

These diagrams represent the standard test cases used for power flow studies and serve as benchmarks for validating the performance of our GNN-based models.

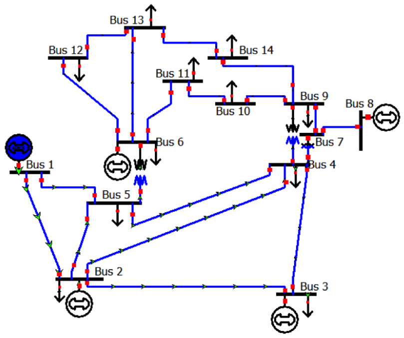

Figure 2: IEEE 14-Bus System

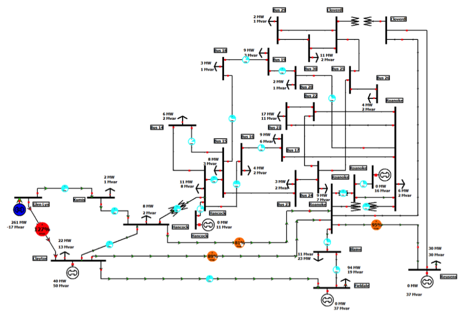

Figure 3: IEEE 30-Bus System

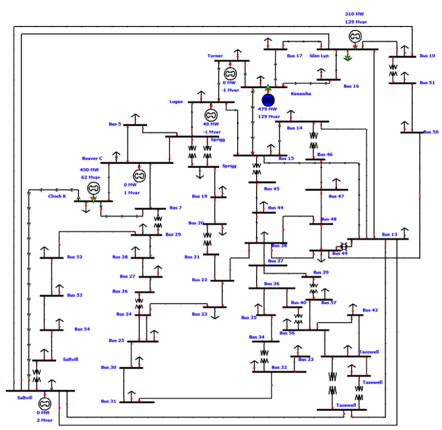

Figure 4: IEEE 57-Bus System

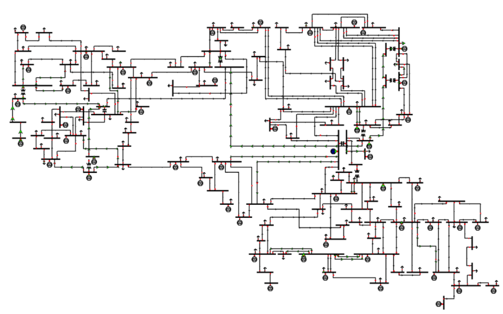

Figure 5: IEEE 118-Bus System

Figure 6: Diagrams of IEEE Bus Systems: 14-Bus, 30-Bus, 57-Bus, and 118-Bus configurations.

Table 1: Dataset Structure for IEEE 30-Bus System

Data Point	**Node**	**Input Features (P, Q, V, δ)**	**Output Features (V, δ)**
Data Point 1	1	P_1, Q_1, V_1, δ_1	V_1, δ_1
	2	P_2, Q_2, V_2, δ_2	V_2, δ_2
	3	P_3, Q_3, V_3, δ_3	V_3, δ_3
	4	P_4, Q_4, V_4, δ_4	V_4, δ_4
	...	...	...
	30	$P_{30}, Q_{30}, V_{30}, \delta_{30}$	V_{30}, δ_{30}
Data Point 10,000	1	P_1, Q_1, V_1, δ_1	V_1, δ_1
	2	P_2, Q_2, V_2, δ_2	V_2, δ_2
	3	P_3, Q_3, V_3, δ_3	V_3, δ_3
	4	P_4, Q_4, V_4, δ_4	V_4, δ_4
	...	...	...
	30	$P_{30}, Q_{30}, V_{30}, \delta_{30}$	V_{30}, δ_{30}

Daily Load Variation Daily load variation is influenced by typical human activity and industrial operations, creating distinct patterns throughout the day.

- **Morning Ramp-Up (6 AM - 9 AM):** Load demand increases as residential and commercial activities begin (60 to 70 of the base load).
- **Midday Peak (10 AM - 3 PM):** Load peaks due to industrial and commercial energy usage (110 to 120 of the base load).
- **Evening Peak (5 PM - 9 PM):** A higher peak occurs as people return home, increasing residential load (110 to 120 of the base load).
- **Nighttime Drop (11 PM - 5 AM):** Load decreases significantly during late night hours (60 to 70 of the base load).

Seasonal Load Variation Seasonal variations also reflect changes in load due to heating and cooling needs, influenced by weather and climate conditions:

- **Winter Peak:** Higher loads due to increased heating demand in colder climates (1.2 to 1.4 times the base load).
- **Summer Peak:** Higher loads due to increased air conditioning usage in hot climates (1.1 to 1.3 times the base load).
- **Spring/Fall Load:** Moderate loads due to milder weather conditions.

These %40 realistic variations ensure that our models are tested under different operating conditions, capturing both high-demand scenarios and low-demand scenarios. This helps in evaluating the model's ability to generalize and maintain performance across a wide range of load conditions.

3.3 Ground Truth Generation

Ground truth data was generated using the Newton-Raphson method to solve the AC Power Flow equations for each load scenario. The Newton-Raphson method was chosen due to its robustness and fast convergence, making it ideal for solving the non-linear power flow equations accurately. We also briefly considered Gauss-Seidel and Fast-Decoupled methods for comparison but found Newton-Raphson to be superior in accuracy and convergence rate.

3.4 GNN Architecture and Topologies to Test

The following topologies will be tested to compare the effectiveness of different GNN architectures in solving the Optimal Power Flow (OPF) problem. Each topology offers distinct strengths, such as handling local versus long-range relationships, incorporating attention mechanisms, and scaling to larger grid sizes:

1. **Graph Convolutional Networks (GCN):** GCN is a basic form of GNN where each node aggregates features from its immediate neighbors. Its simplicity makes it a strong first model to set a benchmark. GCN is widely used for node classification and regression tasks and serves as a baseline model. It can effectively handle local relationships but may struggle with capturing long-range dependencies in the graph.
2. **Graph Attention Networks (GAT):** `GATConv` GAT assigns different attention weights to each neighboring node during the aggregation process, helping the network focus on the most important nodes. This dynamic weighting of neighbors is particularly useful in OPF tasks when certain nodes (such as generator buses) have a greater influence on the power flow solution. The attention mechanism allows the model to prioritize critical nodes, potentially improving prediction accuracy.
3. **GraphSAGE (Sample and Aggregate):** `SAGEConv` GraphSAGE is designed for inductive learning and allows the model to sample a fixed-size neighborhood of nodes, making it scalable to large graphs. GraphSAGE is particularly well-suited for OPF tasks in large power grids, as it can efficiently handle expanding network sizes. It aggregates sampled neighbor features, making it robust for generalizing to unseen nodes or changes in the grid topology.

4. **Graph Convolution (GraphConv):** `GraphConv` GraphConv enhances feature learning by incorporating self-loops, where each node aggregates its own feature along with its neighbors. This topology performs a weighted sum of neighbor features, including the node's own feature, which helps capture more node-specific information. GraphConv is especially useful in scenarios where enhanced node feature aggregation (like generator buses) can lead to better model performance. It is expected to perform well in grids where local node information is crucial.

The selected GNN topologies serve distinct purposes in our experiments:

- **Baseline:** `GCN` serves as the benchmark model to establish a performance baseline for comparison with more complex architectures.
- **Attention-Based:** `GATConv` is used to evaluate whether incorporating attention mechanisms helps improve performance in OPF tasks, particularly by prioritizing influential nodes.
- **Scalability Test:** `SAGEConv` is included to assess the model's scalability and performance on larger bus systems, leveraging its inductive learning capabilities.
- **Enhanced Node Feature Learning:** `GraphConv` is used to investigate whether incorporating self-loops and enhanced node feature aggregation improves the model's performance, especially for nodes like generator buses.

This feature set ensures that the model can differentiate between bus types, which is crucial for accurate power flow predictions and was not explicitly handled in previous baseline models.

Table 2: Summary of Hyperparameters for GNN Models

Hyperparameter Settings	
Hyperparameter	**Value**
Learning Rate	5×10^{-5}
L2 Regularization (Lambda)	1×10^{-6}
Dropout Rate	0.2
GNN Type	GCN / GAT / SAGEConv
Input Features	7 (Voltage, Phase Angle, Active Power, Reactive Power, PV, PQ, Slack Bus Types)
Number of GNN Layers	2
GNN Layer 1 Size	12
GNN Layer 2 Size	12
Hidden Layer Size (Fully Connected)	128
Output Size	$2 \times n\_bus$ (Voltage Magnitude and Angle for Each Bus)
Batch Size	128
Optimizer	Adam
Learning Rate Scheduler	Exponential Decay (Decay Factor: 0.9 every 10 epochs)
Total Trainable Parameters	Varies by GNN Type
Early Stopping Patience	20 epochs

4 Experimental Setup and Dataset Generation

4.1 Experimental Setup detail

To evaluate the performance of our GNN models, we conducted a comprehensive experimental setup involving dataset generation, normalization, model training, and testing across multiple IEEE test cases (14-bus, 30-bus, and 57-bus systems). The details of each step are described below:

4.1.1 Dataset Preparation

We used pre-processed datasets generated from power flow simulations based on IEEE test systems (14-bus, 30-bus, and 57-bus). The datasets were loaded from Excel files and divided into training, validation, and test sets as follows:

- **Train Dataset:** Consisted of 100% of the first dataset file for each bus system.
- **Validation Dataset:** Consisted of 20% of a separate second dataset file to ensure an unbiased validation process.
- **Test Dataset:** For each bus system, we used 20% of 10 separate test datasets, evaluating the model on diverse load scenarios.

4.1.2 Feature Extraction and Dataset Creation

The input features for each bus included the following values:

- Active Power (P), Reactive Power (Q), Voltage Magnitude (V), Voltage Angle (δ).
- Bus Type (Slack, PV, PQ), represented as one-hot encoded features.

We defined the bus type as:

Table 3: Bus Type Definitions and Known/Unknown Variables

Bus Type	**Known Variables**	**Unknown Variables**
Slack Bus	V, δ	P, Q
PV Bus (Generator Bus)	P, V	Q, δ
PQ Bus (Load Bus)	P, Q	V, δ

The target outputs were the voltage magnitude (V) and voltage angle (δ) for each bus.

4.1.3 Data Normalization

The features and targets were normalized to improve model training stability. We computed the mean and standard deviation for both input features and targets. All features except the bus type were normalized using z-score normalization. The bus type was kept as categorical and not normalized. The denormalization process was applied to predictions during evaluation to obtain actual voltage magnitudes and angles.

4.1.4 Graph Construction and Data Loaders

The bus systems were represented as graph-structured data using PyTorch Geometric. The edge index was constructed from the 'frombus' and 'tobus' connections in the network topology. Each graph data object consisted of Node Features, Edge Index and Target Outputs. Edge Index Represents bidirectional connections between buses based on the transmission line data. We used PyTorch DataLoaders for batching the graph data with a batch size of 16 for both training and validation.

4.1.5 Training Procedure and Hyperparameter Tuning

The GNN models were trained using the following setup:

- **Optimizer:** Adam optimizer with an initial learning rate of 5×10^{-5} and L2 regularization ($\lambda = 1 \times 10^{-6}$).
- **Learning Rate Scheduler:** We used a ReduceLROnPlateau scheduler to adjust the learning rate based on validation loss improvements.
- **Batch Size:** A batch size of 16 was used for both training and validation phases. Increasing the batch size helped in convergence and noise avoidance.
- **Epochs:** Models were trained for a maximum of 800 epochs with early stopping based on validation loss, using a patience of 100 epochs.
- **Dropout and Batch Normalization:** We used a dropout rate of 0.2 and enabled batch normalization to prevent overfitting and improve model generalization.

4.1.6 Model Evaluation and Testing

The performance of the GNN models was assessed using the following metrics:

- **Mean Squared Error (MSE):** Measures the average squared difference between predicted and true values of voltage magnitude (V) and phase angle (δ).
- **Normalized Root Mean Squared Error (NRMSE):** Provides a normalized measure of error, allowing for comparison across different test cases.
- R^2 **Score:** Indicates the proportion of variance in the target variables explained by the model, with higher scores indicating better predictive performance.
- **Constraint Violation Rate:** Quantifies the percentage of predictions violating physical constraints, such as voltage bounds and generator capacity limits.

For each test dataset (10 different scenarios for each bus system), we loaded the data and prepared it for evaluation. The best-performing model (based on validation loss) was used for testing. Predictions were denormalized, and regression metrics (MSE, RMSE, NRMSE, MAE, R^2) were calculated. The test losses were saved and summarized across all datasets, providing a comprehensive assessment of model performance.

4.2 Results and Preliminary Analysis

We plotted the training and validation loss curves for the IEEE 14-bus, 30-bus, and 57-bus systems to visualize the model's convergence. The loss curves demonstrate consistent convergence across all test cases, indicating effective learning and generalization. Additionally, we compared the performance of different GNN architectures (GCN, GAT, and SAGEConv) using three evaluation metrics: NRMSE, R^2 score, and average training loss.

The consistent trend between training and validation losses suggests effective learning and good generalization.

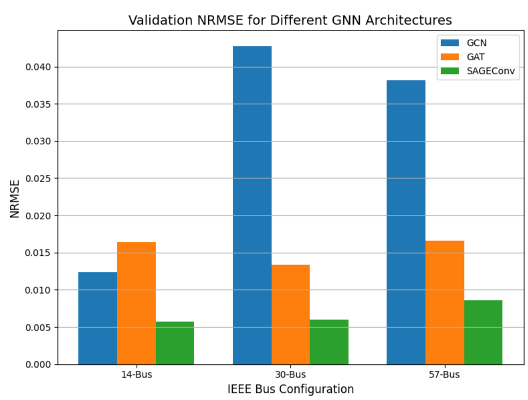

Figure 7: Comparison of NRMSE across different IEEE bus configurations (14-bus, 30-bus, and 57-bus) using three GNN architectures: GCN, GAT, and SAGEConv.

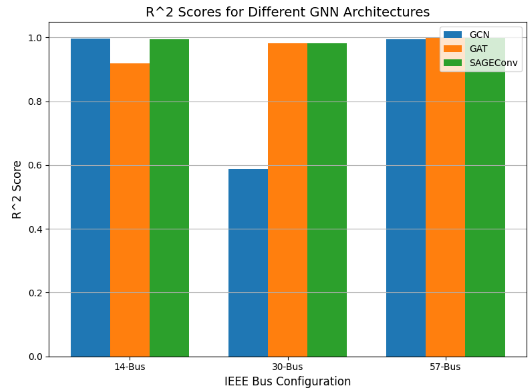

Figure 8: Comparison of R^2 scores for different IEEE bus configurations (14-bus, 30-bus, and 57-bus) using GCN, GAT, and SAGEConv.

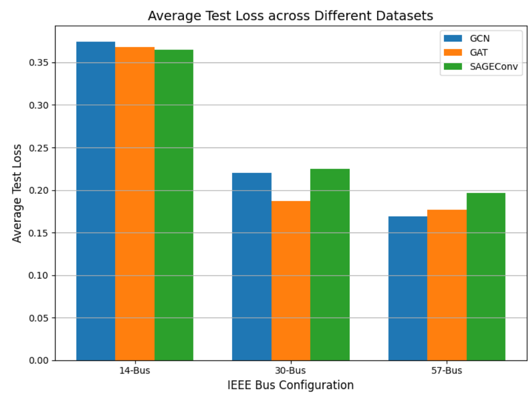

Figure 9: Comparison of average test loss across different IEEE bus configurations (14-bus, 30-bus, and 57-bus) using GCN, GAT, and SAGEConv.

Analysis: The results highlight the strengths and weaknesses of each GNN architecture across different bus systems. Also it is noted that as the number of buses decrease the training loss also decrease. The models achieved consistent performance across all test cases, with NRMSE values below 0.05. Increasing the batch size and reducing the dropout rate led to improved convergence and lower test loss. The findings suggest that attention-based models like GAT and scalable models like SAGEConv are promising choices for power flow prediction tasks, particularly in larger and more complex grid scenarios. SAGEConv's ability to handle large datasets effectively, combined with GAT's focus on critical nodes, makes them suitable for further exploration and potential applications in real-world power systems.

- **GCN:** GCN showed higher NRMSE values across all bus configurations, particularly for the 30-bus and 57-bus systems, indicating challenges in capturing complex and long-range dependencies. The R^2 scores for GCN were generally lower compared to GAT and SAGEConv which suggests limited model fit, especially for larger grids. Average training loss for GCN was higher which imply slower convergence.
- **GAT:** GAT demonstrated significant improvements in NRMSE, especially for the 14-bus and 30-bus systems which leverage its attention mechanism to focus on critical nodes (e.g., generator buses). The R^2 scores for GAT were consistently high which indicate a strong fit to the target data, especially on the 30-bus and 57-bus systems. GAT achieved the lowest average training loss which shows its efficiency in learning meaningful representations, particularly for smaller and medium-sized grids.
- **SAGEConv:** SAGEConv consistently achieved the lowest NRMSE values across all bus configurations, demonstrating its scalability and superior performance in handling larger

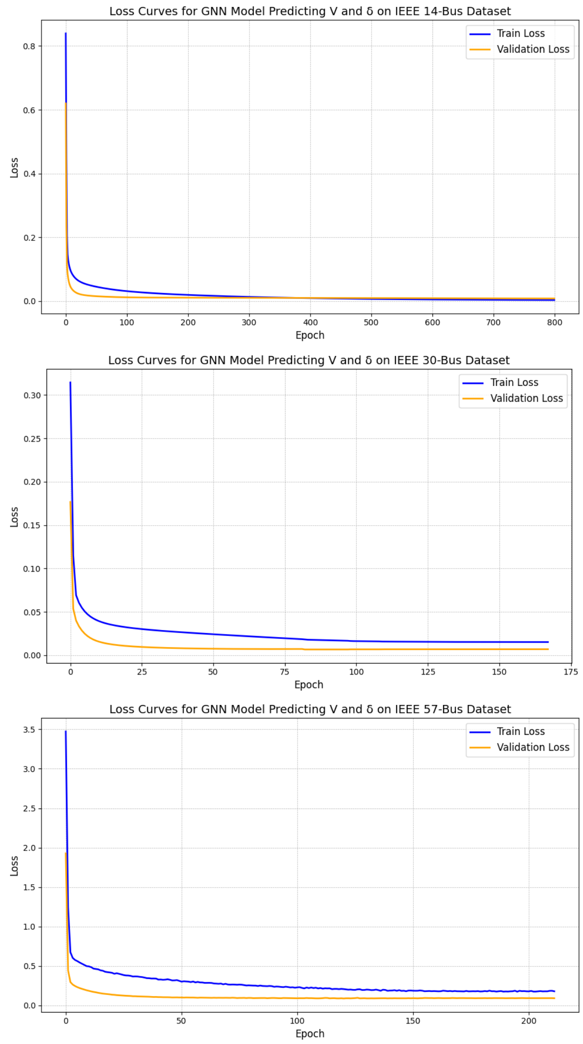

Figure 10: Training and validation loss curves for the IEEE 14-bus, 30-bus, and 57-bus systems using GCN Architecture.

systems like the 57-bus network. R^2 scores for SAGEConv were comparable to those of GAT, showing excellent model fit and robustness, especially for the largest test case (57-bus). Despite slightly higher training loss compared to GAT, SAGEConv showed strong generalization, as evidenced by its lower NRMSE and high R^2 scores across all configurations.

4.3 Future Work

While our initial results are promising, further testing on the IEEE 118-bus system is planned to evaluate the model's scalability. We aim to explore additional GNN architectures for improved performance.

Conclusion

This paper presents a novel GNN-based approach for solving the AC Power Flow problem, incorporating voltage magnitude, voltage angle updates, and bus type as features—a capability not addressed by baseline methods. Preliminary results on IEEE test cases demonstrate strong performance, and future work will focus on extending this approach to larger systems and enhancing constraint handling.

References

[1] M. B. Cain, R. P. O'Neill, and A. Castillo, "History of optimal power flow and formulations," Federal Energy Regulatory Commission, Increasing Efficiency through Improved Software, pp. 1–31, Dec. 2012

[2] M. S. Vijayabaskar, V. Niranjan, and S. Vishveshwara. Graprostr - graphs of protein structures: A tool for constructing the graphs and generating graph parameters for protein structures. Open Bioinformatics Journal, 5:53–58, Feb. 2011.

[3] I. Ahmad, M. Akhtar, S. Noor, and A. Shahnaz. Missing link prediction using common neighbor and centrality based parameterized algorithm. Scientific Reports, 10:364, 01 2020. doi: 10.1038/s41598-019-57304-y

[4] S. Hosein, P. Hosein, W. Kattick, and V. Ratan. Web application for power grid fault management. pp. 1–5, 08 2016. doi: 10.1109/ICIAS.2016.7824052

[5] N. Guha, Z. Wang, and A. Majumdar, "Machine learning for AC optimal power flow," in 36th Int. Conf. Mach. Learning, Long Beach, CA, 9-15 June 2019.

[6] Chen, Z.,& Yu, C. (2020). Optimal Power Flow Using Graph Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence.

[7] Donti, P. L., Amos, B., & Kolter, J. Z. (2021). Topology-Aware Graph Neural Networks for Learning Feasible and Adaptive AC-OPF Solutions. IEEE Transactions on Power Systems, 36(1), 258–270.

[8] López-García, T. B., & Domínguez-Navarro, J. A. (2023). Power Flow Analysis via Typed Graph Neural Networks. Engineering Applications of Artificial Intelligence, 117, 105567.

[9] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang and Y. Wang, "A Review of Graph Neural Networks and Their Applications in Power Systems," in Journal of Modern Power Systems and Clean Energy, vol. 10, no. 2, pp. 345-360, March 2022, doi: 10.35833/MPCE.2021.000058.

[10] Yang, Y., Wu, H., & He, H. (2020). Proximal Policy Optimization with Graph Neural Networks for Optimal Power Flow. IEEE Transactions on Power Systems, 35(1), 731–740.

[11] Donon, B., Clément, R., Donnot, B., Marot, A., & Guyon, I. (2020). Neural Networks for Power Flow: Graph Neural Solver. Electric Power Systems Research, 189, 106547.

[12] Fan, W., Li, X., & Wang, B. (2021). Graph Neural Network and Koopman Models for Learning Networked Dynamics: A Comparative Study on Power Grid Transients Prediction. IEEE Transactions on Circuits and Systems I: Regular Papers, 68(10), 4208–4221.

[13] A. B. Jeddi and A. Shafieezadeh, "A Physics-Informed Graph Attention-based Approach for Power Flow Analysis," 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 2021, pp. 1634-1640, doi: 10.1109/ICMLA52953.2021.00261.

[14] H. Wu, M. Wang, Z. Xu and Y. Jia, "Graph Attention Enabled Convolutional Network for Distribution System Probabilistic Power Flow," in IEEE Transactions on Industry Applications, vol. 58, no. 6, pp. 7068-7078, Nov.-Dec. 2022, doi: 10.1109/TIA.2022.3202159.